VŠB – Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Systém pro detekci zneužití SW PBX Asterisk

System for Fraud Detection for Asterisk SW PBX

2013

Aleš Procházka

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání bakalářské práce

Student: **Aleš Procházka**

Studijní program: B2647 Informační a komunikační technologie

Studijní obor: 2612R025 Informatika a výpočetní technika

Téma: Systém pro detekci zneužití SW PBX Asterisk
System for Fraud Detection for Asterisk SW PBX

Zásady pro vypracování:

Cílem této práce je navrhnout a implementovat software pro detekci zneužití telefonních ústředen na bázi SW Asterisk.
Práce bude obsahovat:
1. Obeznámení a shrnutí problematiky detekce zneužití (fraud detection) ústředen na bázi SW Asterisk.
2. Zdokumentování možnosti logování hovorů do SQL DBMS a práci s AMI (asterisk management interface) a CEL (channel event logging).
3. Návrh a implementace softwaru pro detekci zneužití ústředny sledováním objemu a typů hovorů.

Seznam doporučené odborné literatury:

Podle pokynů vedoucího bakalářské práce.

Formální náležitosti a rozsah bakalářské práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí bakalářské práce: **Ing. Nikola Ciprich**

Datum zadání: 16.11.2012

Datum odevzdání: 07.05.2013

VYSOKÁ ŠKOLA BÁŇSKÁ-TECHNICKÁ UNIVERZITA Fakulta elektrotechniky a informatiky OSTRAVA

doc. Dr. Ing. Eduard Sojka
vedoucí katedry

prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Prohlašuji, že jsem tuto bakalářskou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Uherském Hradišti 22. dubna 2013

.....

Rád bych na tomto místě poděkoval mému vedoucímu bakalářské práce Ing. Nikolu Ciprichovi, za velmi užitečnou metodickou pomoc a za čas, který věnoval mé bakalářské práci.

Abstrakt

Tato bakalářská práce je zaměřena na detekci zneužití telefonních ústředen na bázi SW Asterisk. V práci bude popsána VoIP ústředna Asterisk a její formy zneužití, zdokumentovány možnosti logování hovorů do DBMS a předvedena základní práce s AMI. Zvláštní pozornost bude věnována navržení a implementaci softwaru pro detekci zneužití ústředny sledováním objemu a typů hovorů. V závěru této práce je software předveden na reálných útocích.

Klíčová slova: Asterisk, detekce zneužití, logování hovorů, CDR, CEL, AMI

Abstract

This bachelor thesis is focused on fraud detection of telephone exchanges on Asterisk-based software. In the thesis will be described VoIP PBX Asterisk and its forms of frauds, documented logging options calls to database and demonstrated basic work with AMI. Particular attention will be paid to the design and implementation of software to fraud detect on PBX by monitoring the volume and types of calls. In the conclusion of this thesis the software will present on real attacks.

Keywords: Asterisk, Fraud Detection, Call Logging, CDR, CEL, AMI

Seznam použitých zkratek a symbolů

AMA	–	Automatic Message Accounting
AMI	–	Asterisk Manager Interface
ANI	–	Automatic number identification
CDR	–	Call Detail Records
CEL	–	Channel Event Logging
CELT	–	Constrained Energy Lapped Transform
CLI	–	Command-line interface
CSV	–	Comma-separated values
DBMS	–	Database management system
DDoS	–	Distributed Denial of Service
DoS	–	Denial of Service
GPL	–	General Public License
GSM	–	Global System for Mobile Communications
GUI	–	Graphical User Interface
HTTP	–	Hypertext Transfer Protocol
IAX	–	Inter-Asterisk eXchange
IETF	–	Internet Engineering Task Force
iLBC	–	Internet Low Bitrate Codec
IP	–	Internet Protocol
ITU	–	International Telecommunication Union
IVR	–	Interactive voice response
MGCP	–	Media Gateway Control Protocol
NAT	–	Network Address Translation
ODBC	–	Open Database Connectivity
PBX	–	Private branch exchange
PL/pgSQL	–	Procedural Language/PostgreSQL
PSTN	–	Public switched telephone network
RDBMS	–	Relational database management system
RFC	–	Request for Comments
RTP	–	Real-time Transport Protocol
SCCP	–	Skinny Client Control Protocol
SCIP	–	Secure Communications Interoperability Protocol

SDP	–	Session Description Protocol
SGCP	–	Simple Gateway Control Protocol
SIP	–	Session Initiation Protocol
SMTP	–	Simple Mail Transfer Protocol
SPIT	–	Spam over Internet telephony
TCP	–	Transmission Control Protocol
TDM	–	Time-division multiplexing
UDP	–	User Datagram Protocol
UNISTIM	–	Unified Networks IP Stimulus
VoIP	–	Voice over Internet Protocol

Obsah

Seznam tabulek

Seznam obrázků

Seznam výpisů zdrojového kódu

1 Úvod

Historie VoIP (Voice over Internet Protocol) sahá až do doby před vznikem Internetu. První pokusy uskutečnit VoIP hovory se objevily již v roce 1973. V roce 1995 byl vytvořen první počítačový program umožňující hovor přes IP (Internet Protocol), který znamenal velký rozmach VoIP technologie. Ten pravý „boom“ však nastal až s nástupem širokopásmových připojení. Nyní patří VoIP již k běžně používané technologii, a to z důvodu dostupnosti a přijatelné ceně hardwaru a softwaru. Velkou výhodou VoIP je úspora nákladů za hovory a nulové poplatky za vedení linky. Nevýhodou je velká závislost na kvalitě linky a také nižší bezpečnost. VoIP je oproti klasickým PSTN více zranitelnější, což spolu s rozmachem VoIP technologie otevřelo nový prostor pro kriminální činy, které se pro mnohé staly výnosným byznysem.

V této bakalářské práci bude nejdříve představen SW Asterisk, který je dnes nejpoužívanějším open source softwarem implementujícím telefonní ústřednu (PBX). Bude nastíněna historie, kodeky, protokoly, nejčastější možnosti užití a základní instalace tohoto softwaru. Dále budou rozebrány základní formy zneužití, bezpečnostní hrozby a bezpečnostní opatření umožňující eliminovat rizika spojená s provozováním SW Asterisk. V práci budou zdokumentovány možnosti a základní konfigurace logování hovorů do DBMS formou CDR (Call Detail Records) a CEL (Channel Event Logging). Dále bude popsána základní práce s AMI (Asterisk Manager Interface), včetně možnosti ukončení aktivního hovoru. Dalším cílem bakalářské práce bude navržení a implementace softwaru pro detekci zneužití ústředny sledováním objemu a typů hovorů. Software bude obsahovat tarifikační systém a systém na samotnou detekci zneužitých účtů, který bude na zneužití upozorňovat formou alarmů. Následně bude tento software otestován na reálných útocích.

Konfigurace budou povedeny na linuxové distribuci Debian GNU/Linux 6.0 s verzí Asterisk 1.8.11. Jako DBMS bude použit PostgreSQL 8.4 s využitím procedurálního jazyka PL/pgSQL.

2 Co je to Asterisk

”Oficiálně je Asterisk open source hybrid TDM a packet voice PBX, jedná se o IVR (Interactive Voice Response) platformu s funkčností Automatic Call Distribution (ACD). Neoficiálně jde možná o jedno z „nejsilnějších“, flexibilních a rozšířitelných řešení v oblasti integrovaného telekomunikačního softwaru.”[5] Asterisk je open-source s GNU licencí sponzorovaný společností Digium. Dokáže běžet na Linuxu, OS X, BSD a emulovaně i na Windows, nicméně je navržen tak, aby pracoval na Linuxovém jádře.[5]

V této kapitole proběhne seznámení s SW Asterisk, jeho historií, podporovanými kodeky a protokoly. Dále budou uvedeny základní možnosti využití tohoto softwaru a v poslední částí této kapitoly bude naznačena základní instalace na Linuxu.

2.1 Historie Asterisku

V roce 1999 Mark Spencer dokončil studium výpočetní techniky na univerzitě v Auburnu, kde přišel na zajímavý způsob podnikání. V roce 1999 tisíce podniků po celém světě nacházelo nový způsob, jak ušetřit peníze přechodem z proprietárních systémů na open source operační systém Linux. V té době existovalo málo firem, které by se zabývaly podporou pro uživatele Linuxu. Mark se rozhodl vyplnit tuto díru na trhu a založil společnost „Linux Support Services“. Tato společnost nabízela IT odborníkům placenou podporu operačního systému Linux. Za několik měsíců měl Mark malou kancelář obsazenou odborníky a firma rostla tak rychle, že Mark potřeboval novou telefonní ústřednu. Obrátil se na místní prodejce telefonních ústředen, ale nabízená cena byla pro Markovu společnost příliš vysoká. Namísto žádosti o úvěr se Mark rozhodl vytvořit vlastní telefonní systém. O pár měsíců později vydal zdrojový kód funkčního prototypu na Internet, který byl dostupný pod licencí GPL. Tato myšlenka se velmi rychle ujala. Během toho, co Mark přidával nové verze jádra Asterisku, stovky vývojářů přidávalo nové funkce a vlastnosti. Postupem času bylo vytvořeno několik podobných open source projektů, ale žádný z nich nedosáhl takového úspěchu jako Asterisk. V současné době existuje již více než milion komunikačních serverů na bázi Asterisku.[1]

2.2 Podporované kodeky

Kodek je zařízení nebo počítačový program, který převádí analogový hlas na digitální signál, který umožňuje hlasu projít přes Internet. Asterisk podporuje širokou škálu licencovaných či nelicencovaných kodeků, které využívají různou šířku pásma. Na základě výběru kodeku se pak odvíjí kvalita probíhajícího hovoru. Přehled podporovaných kodeků je uveden v Tabulce 1.

Kodek	Šířka pásma
G.711 u-law,a-law[1]	64 kbit/s
G.719	128 kbit/s
G.722	64 kbit/s
G.722.1	16,24,32 kbit/s
G.723.1	5.3 a 6.3 kbit/s
G.726	32 kbit/s
G.729	8 kbit/s
GSM	13kbit/s
iLBC	15,2 kbit/s pro rámec 20 ms a 13.33 kbit/s pro 30 ms
LPC-10	2,4kbit/s
Speex	2 kbit/s až 44 kbit/s
CELT	24 až 128 kbit/s
Silk	6 až 40 kbit/s

Tabulka 1: Podporované kodeky

2.3 Podporované protokoly

Pro provedení spojení VoIP mezi jednotlivými koncovými body je potřeba provést řadu signalizačních transakcí, které vytvoří datové toky nesoucí skutečnou konverzaci (pro každý směr jeden).[2] V této části bude nastíněn základní popis těchto protokolů:

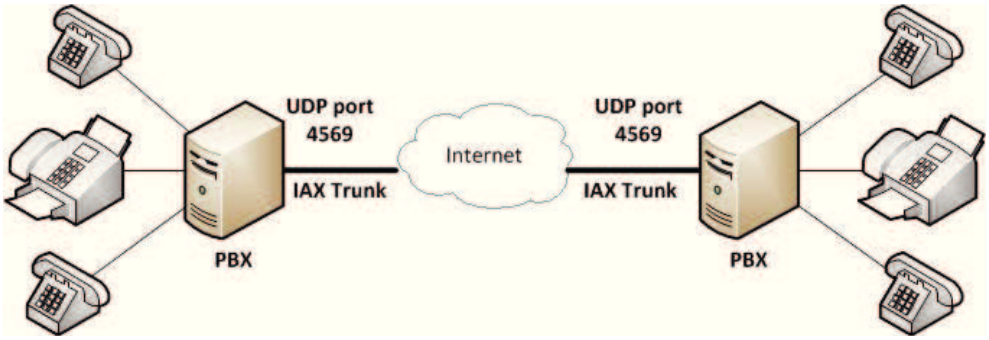
- IAX
- SIP
- H.323
- MGCP
- Skinny/SCCP
- UNISTIM

2.3.1 IAX (Inter-Asterisk eXchange)

"Tvůrcem protokolu IAX stejně jako SW open-source ústředny Asterisk je Mark Spencer."[5] IAX je otevřený protokol aplikační vrstvy určený pro signalizaci a přenos média způsobem peer-to-peer, který byl vyvinut za účelem komunikace s ostatními Asterisk servery, a také jako alternativa k protokolům SIP a H.323. IAX se odlišuje od ostatních protokolů tím, že umožňuje více relací sloučit do tzv. „trunk relace“, čímž dochází ke snížení režie spojené s jednotlivými kanály. To

[1] u-law je logická komprese hovorového signálu používaná v severní Americe a Japonsku a a-law je používaná v Evropě a Austrálii

umožňuje snížení využívané šířky pásma a latence. Signalizace a přenos média jsou realizovány jedním datovým tokem prostřednictvím UDP portu 4569, což vytváří jeho další nepochybnou výhodou, kterou je schopnost projít přes NAT bez jakékoliv další konfigurace. V současné době je v Asterisku IAX ve verzi 2 podporován modulem `chan_iax2.so`. [5, 2]



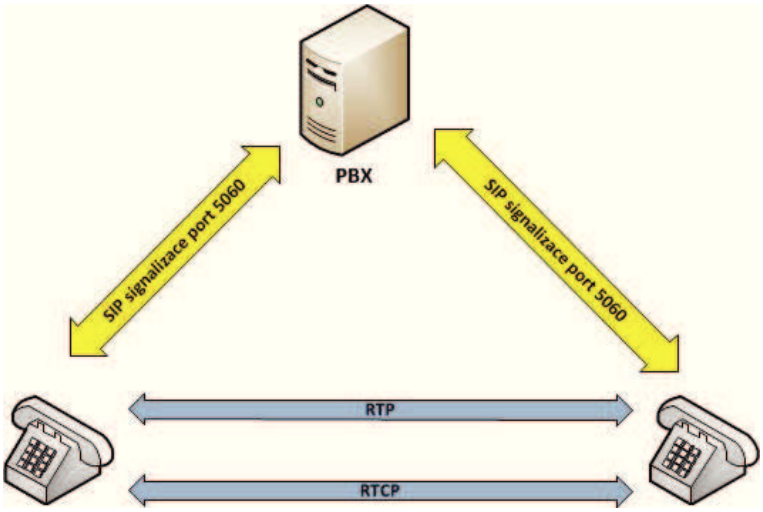
Obrázek 1: Protokol IAX

2.3.2 SIP (Session Initiation Protocol)

SIP "je jednoduchý obecný protokol pro navazování interaktivních komunikačních relací mezi dvěma nebo více koncovými zařízeními v Internetu, které si vyhledá."[6] SIP byl poprvé předložen IETF (Internet Engineering Task Force) v roce 1996 kdy v sobě nesl prvky protokolu SCIP a zcela se lišil od SIP využívaného v dnešní době. V březnu 1999 se po 11 revizích zrodil SIP RFC 2543, současně využívanou druhou verzi popisuje RFC 3261. SIP je signalizační protokol aplikační vrstvy a lze jej přenášet přes UDP i TCP na portu 5060. Jedná se o textový protokol, který je strukturou podobný protokolům HTTP nebo SMTP a jeho tělo obsahuje textové položky ve formě `<název>:<hodnota>`. Textový základ protokolu napomáhá jednoduchému ladění, ale také snadné rozšiřitelnosti. Jedná se o protokol typu server-klient. Jak již bylo zmíněno, SIP je pouze signalizační protokol. Aby byl uskutečněn přenos hlasu, tak musí spolupracovat s protokoly SDP (Session Description Protocol) a RTP (Real-time Transport Protocol). Signalizace SIP a přenos hlasu přes protokol RTP, jak uvádí Obrázek č.2, jsou obvykle realizovány ve tvaru lichoběžníku. Podporu protokolu SIP v Asterisku poskytuje modul `chan_sip.so`.[2, 7, 3]

SIP poskytuje níže uvedené služby:

- **Lokalizace účastníka** – nalezení koncové stanice pro danou komunikaci
- **Navázání spojení** - stanovení parametrů pro obě strany
- **Zjištění stavu účastníka** – zjištění dostupnosti účastníka (zda není obsazeno)
- **Zjištění možností účastníka** – určení přenosové rychlosti, kodeků atd.



Obrázek 2: Protokol SIP (canreinvite=yes)

2.3.3 H.323

H.323 představuje mezinárodní standard ITU VoIP protokolu a byl vytvořen přibližně ve stejnou dobu jako protokol SIP. První verze byla uveřejněna v roce 1996 jako nástroj pro přenos hlasu, videa, dat a faxu přes protokol IP. H.323 není v Asterisku příliš využívaný z důvodu preferování protokolů SIP a IAX. Dalším faktorem, který protokol činí nepopulárním, je jeho složitost. Nicméně Asterisk podporuje tři verze protokolu H.323 pomocí modulů `chan_h323.so`, `chan_oh323.so` a `chan_ooh323.so`.[2]

2.3.4 MGCP (Media Gateway Control Protocol)

Signalizační protokol MGCP, stejně jako protokol SIP, vzešel z IETF a je definován v RFC 3435. Je nástupcem protokolu SGCP a používá se k řízení Media Gateways (MG) na IP sítích a veřejných telefonních sítí (PSTN). Asterisk podporuje MGCP za pomoci modulu `chan_mgcp.so`, přičemž definování koncových bodů probíhá v konfiguračním souboru `mgcp.conf`.[2]

2.3.5 Skinny/SCCP (Skinny Client Control Protocol)

Skinny/SCCP je proprietární protokol ve vlastnictví firmy Cisco, přesto jej Asterisk podporuje. SCCP je standardním protokolem, který se používá pro koncové body na ústředně Cisco Call Manager. [2]

2.3.6 UNISTIM

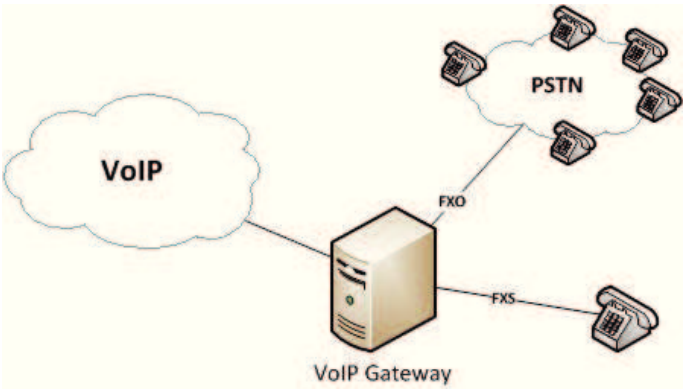
UNISTIM je proprietární protokol ve vlastnictví společnosti Nortel. Podpora tohoto protokolu Asteriskem je zatím experimentální, nicméně Asterisk je historicky první telefonní ústřednou, která podporuje proprietární protokoly dvou největších hráčů na trhu s VoIP (Cisco a Nortel). [2]

2.4 Možnosti využití SW Asterisk

Díky svým funkcím nabízí Asterisk širokou škálu využití. S novými verzemi Asterisku přibývají funkce a rozšiřuje se tím spektrum jeho využití. V následujících kapitolách bude uvedeno několik možností využití SW Asterisk.

2.4.1 VoIP brána (SIP,IAX,H.323,MGCP)

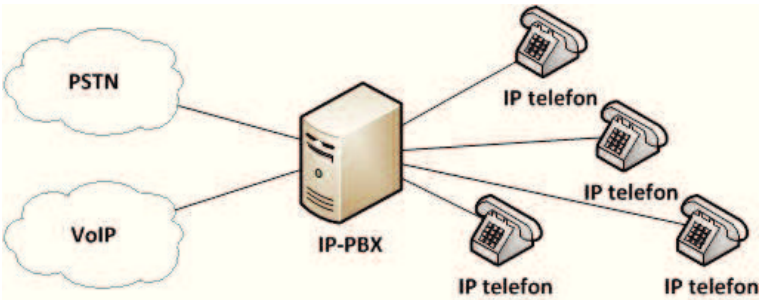
VoIP brána je tzv. most mezi digitální a analogovou telefonií. Umožňuje bezproblémové propojení, přenos hlasu a signalizace mezi těmito dvěma různými infrastrukturami, které pracují s odlišnými standardy a protokoly. (viz. Obrázek č.3)



Obrázek 3: VoIP brána

2.4.2 Pobočková ústředna (PBX)

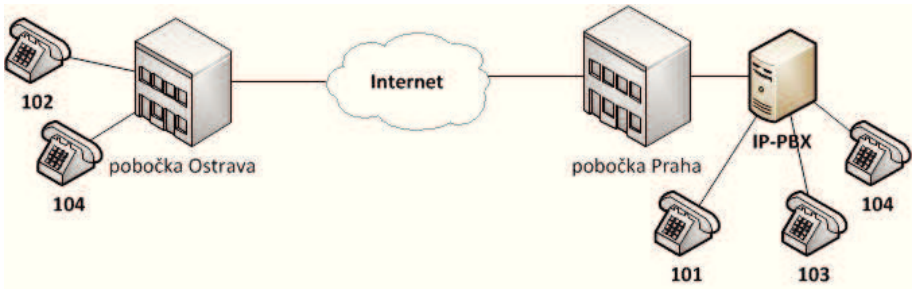
Pobočková ústředna spojuje výstupní body všech firemních telefonů do PSTN. Toto řešení snižuje náklady, protože se neplatí poplatky za telefonní linku pro všechny telefony, ale pouze pro výstupní bod z PBX.(viz. Obrázek č.4)



Obrázek 4: Pobočková ústředna

2.4.3 Propojení vzdálených kanceláří přes jednu PBX

Jestliže má firma několik poboček, které jsou od sebe vzdáleny, lze pomocí Asterisku vytvořit ze všech poboček pouze jedinou, alespoň co se týká telefonní sítě. Toto řešení umožňuje snížit jak náklady na nákup více telefonních ústředen, tak i na telefonních poplatcích. (viz. Obrázek č.5)



Obrázek 5: Propojení vzdálených kanceláří přes jednu PBX

2.4.4 IVR server

IVR je zkratkou pro Interactive Voice Response (interaktivní hlasový průvodce). Jedná se o automatickou interakci mezi volajícím a telefonní ústřednou, která mu zpravidla poskytuje určité služby. Volající komunikuje pomocí tónové volby nebo hlasem. Typickým příkladem IVR je automatický operátor nebo hlasové menu, kde reagujeme výběrem číslice. Toto řešení eliminuje potřebu živého operátora k zajištění konkrétní služby.[4]

2.5 Instalace Asterisku

Asterisk je schopen fungovat na mnoha operačních systémech, ale díky tomu, že byl navržen pro Linux, dokáže nejlépe pracovat právě na tomto systému. Přestože existuje velké množství Linuxových distribucí, tato kapitola bude zaměřena na instalaci Asterisku na jedné z nejpoužívanějších distribucí, a to Debian GNU/Linux, na kterém je založena i další oblíbená distribuce Ubuntu.

2.5.1 Hardwarové požadavky

V první řadě je potřeba zjistit, zda hardware, na který se bude instalovat server, je plně kompatibilní se systémem GNU/Linux. Následně musíme zvážit objem hovorů, které bude server zpracovávat a na základě toho zvolit dostatečně výkonný hardware, protože Asterisk zpracovává požadavky v reálném čase. Podrobněji se tomu věnuje literatura[10].

2.5.2 Instalace Debian

Nejdříve je potřeba nainstalovat Debian GNU/Linux 6.0[2] , který je současnou stabilní verzí. Podrobný návod na instalaci je uveden v literatuře [8].

2.5.3 Instalace Asterisku na server

Nyní se budu zabývat instalací samotného Asterisku. Asterisk vychází ve verzích standard a LTS (Long Term Support). V případě instalace ústředny na delší dobu je dobré zvolit verzi, která nabízí dlouhodobější podporu, tedy LTS. Přehled současných a plánovaných verzí je uveden v literatuře [9].

Před samotnou instalací je potřeba nainstalovat balíky, které umožní správnou kompilaci.

```
$ apt-get install gcc g++ libxml2-dev libncurses5-dev
```

Nyní můžeme stáhnout balíček jedné z verzí Asterisku, která je dostupná na stránkách samotného projektu, [11] a po jeho rozbalení provedeme kompilaci a instalaci.

```
$ ./configure
$ make
$ make install
$ make samples
```

Dále musíme vytvořit initscript a to z toho důvodu, aby po restartu serveru došlo k automatickému spuštění Asterisku.

```
$ make config
```

Tímto je základní instalace hotová. Více uvádí literatura [12].

[2] je možno zvolit i starší verzi Debian GNU/Linux

3 Zneužití ústředen na bázi SW Asterisk

S rostoucí popularitou VoIP roste i počet kriminálních činů soustředících se na tuto technologii. Nebylo možné předpokládat, že stále populárnější technologie VoIP se nestane vyhledávaným kriminálním cílem. Postupem času se útoky na VoIP servery a PBX stávaly častějšími a dokonalejšími. Automatické skenování a hledání bezpečnostních děr probíhá na většině serverů prakticky denně. Jelikož útoky přicházejí z různých IP adres, stávají se zákeřnějšími. Útoky mohou také probíhat ze záměrně podvrhnuté IP adresy, kdy je obtížné zjistit skutečnou IP adresu, která se maskuje za náhodně generovaný shluk IP adres. Tyto útoky mohou představovat pro správce ústředny nemalé problémy. V souvislosti z výše uvedeným představím základní hrozby a rizika spojená s provozováním ústředny Asterisk a také možnosti, jak se před útoky bránit.[26]

3.1 Formy zneužití

Mezi základní formy zneužití PBX patří:

- Toll fraud
- Nevyžádané hlasové zprávy (SPIT)
- Phishing

3.1.1 Toll fraud

U tohoto typu zneužití se útočníci snaží najít slabé místo v systému, které jim umožní volání zdarma na velké vzdálenosti nebo vysoce zpoplatněná čísla. Hovory ve většině případů směřují do exotických destinací, kde cena hovoru za minutu je dosti vysoká. Převážně se jedná o hovory do afrických států, ale i do Severní Korei nebo Číny. Po napadení serveru se útočník snaží o prodej ukradeného hovorného nic netušícím koncovým zákazníkům, což mu umožní nemalý příjem. Tento způsob podvodu představuje pro Asterisk ústředny největší hrozbu proto, že při nich útočník způsobí velkou finanční škodu.[26]

3.1.2 Nevyžádané hlasové zprávy (SPIT)

Nevyžádané hlasové zprávy (spam over Internet telephony) se vyznačují hromadným zasíláním hlasových zpráv přes VoIP protokol na ostatní telefony připojené k Internetu. Většinou se jedná o reklamní sdělení nebo žertovné zprávy. Díky své popularitě a široké podpoře výrobců patří k nejzranitelnějším protokol SIP. Cílem útočníka je umístit na ústřednu tzv. spambota nebo získat telefonní seznam. V současné době tento způsob zneužití ústředny nepatří k příliš častým.[13]

3.1.3 Phishing

Phisning přes VoIP se stává stále populárnějším z toho důvodu, že VoIP je levnější a zranitelnější než klasická PSTN. U VoIP lze manipulovat s ID volajícího, čímž se nabízí útočníkovi možnost vystupovat za důvěryhodné organizace (banky, úřady, atd.), prostřednictvím kterých bude získávat citlivá data oběti.[14]

3.2 Bezpečnostní hrozby

Mezi největší bezpečnostní hrozby, které útočníkovi umožní proniknout do systému a poté zneužít PBX k páchání zločinu, patří:

- Síťové útoky
- Chyby v konfiguraci
- Bezpečnostní chyby v Asterisku
- Lidský faktor

3.2.1 Síťové útoky

Největší bezpečnostní hrozbou pro Asterisk jsou bezesporu síťové útoky. Tyto útoky patří k velmi častým a využívají chyby a slabiny v systému, pomocí kterých se snaží systém zahltit, získat hesla atd. Mezi nejčastější typy útoků patří:

- DoS útok
- DDoS útok
- Skenování portů
- Hádání hesla (Password Guessing)

3.2.2 Chyby v konfiguraci

Chyby v konfiguraci Asterisku mohou zásadně ovlivnit jeho bezpečnost, a tím i možnost jeho zneužití. Mezi základní chyby patří slabá nebo defaultní hesla, která útočníkovi značně usnadní přístup do systému. Dalším rizikem mohou být chyby v dialplanu a dalších konfiguračních souborech, které umožní například anonymní hovory nebo dovolí potenciálnímu útočníkovi volat kamkoliv.[15]

3.2.3 Bezpečnostní chyby v Asterisku

S vydáváním nových verzí Asterisku se můžou objevit bezpečnostní chyby, v důsledku kterých by mohlo dojít ke zneužití ústředny útočníkem.

3.2.4 Lidský faktor

Lidský faktor je zranitelné místo každého systému a Asterisk není výjimkou. Rizikem je například odcizení zařízení (notebook, telefon, atd.), ve kterém je nastaven SIP účet.

3.3 Jak se bránit?

Riziko napadení a zneužití PBX lze snížit dodržením bezpečnostních pravidel a opatření, která by se měla stát do určité míry standardem při konfiguraci ústředny. Zde je uveden základní přehled opatření:

- Firewall
- Silná hesla
- Ochrana před hádáním hesel
- Nastavení limitů hovorů
- Monitoring systému

3.3.1 Firewall

Použití firewallu patří k jednomu z nejzákladnějších bezpečnostních opatření, obzvláště pokud je ústředna dostupná z Internetu. Základní politikou firewallu je povolení pouze těch spojení, která jsou bezpodmínečně nutná pro správnou funkci systému. Zejména se jedná o zakázání nepotřebných služeb a omezení vzdáleného přístupu k databázím. Pokud je Asterisk provozován na Linuxu, můžeme firewall nastavit pomocí `iptables`. Konfigurace linuxového firewallu je dost komplexní záležitost mimo rozsah této práce, podrobněji je řešena v literatuře[17] a [18].[16]

3.3.2 Silná hesla

Mezi další důležitá bezpečnostní opatření patří použití silného hesla. Heslo by mělo být dostatečně dlouhé a obsahovat minimálně 8 znaků. Mělo by být složeno z kombinace malých písmen, velkých písmen, číslic a dalších znaků, přičemž by se nemělo jednat o známá slova, která bývají velmi často využívána ke slovníkovým útokům.

3.3.3 Ochrana před hádáním hesel

V některých případech není zcela jasné, z jaké IP adresy se klienti budou připojovat. Pokud například budou klienti telefonovat z veřejných Wi-Fi nebo 3G sítí, nelze přesně definovat rozsah přípustných IP adres. V tomto případě není možné provést zabezpečení jen za pomoci firewallu a je nutné využít další vrstvu ochrany, která omezuje počet pokusů zadaní hesla pro jednotlivé IP adresy. Na Asterisku máme k dispozici externí nástroj `Fail2Ban`, který tento problém řeší tím, že průběžně sleduje změny v logu (bere v úvahu jen nové záznamy) a po stanoveném množství hlášení o nesprávně zadaném hesle zablokuje danou IP adresu pomocí `iptables`.[16]

3.3.4 Nastavení limitů hovorů

K dalším opatřením bezesporu patří nastavení limitů hovorů pro jednotlivé uživatele. Hovory je možné omezit na maximální provolanou částku za dané období, kdy po překročení limitu bude

každý další hovor zablokován. Dalším opatřením je omezení nebo úplné zablokování hovorů do drahých zahraničních destinací nebo na vysoce zpoplatněné linky. Možné je také omezení maximálního počtu současně sestavených hovorů na jednoho uživatele. Nastavení limitů hovorů Asterisk neumožňuje, ale tento problém lze řešit externím softwarem, kterým se tato bakalářská práce zabývá.

3.3.5 Monitoring systému

Monitorováním a sledováním systému se dá včas odhalit nestandardní chování, a tím předejít pozdějším nežádoucím následkům. Jedna součást monitoringu systému je i analýza CDR nebo CEL záznamů (více o logování hovorů v kapitole 4). Za pomoci této analýzy je možno odhalit zneužití PBX v reálném čase a zabránit tak větším škodám. Analýza by měla zahrnovat i systém, který na základě nestandardního chování, omezí či zablokuje další hovory, a tím eliminuje riziko dalšího zneužití(více v kapitole 6).

4 Možnosti logování hovorů

Asterisk má standardně dva subsystémy, které umožňují získat podrobné informace o právě probíhajících nebo již proběhlých hovorech. Jedná se o subsystémy Call Detail Records a Channel Event Logging. Tyto subsystémy lze využít na sledování objemů hovorů, jako podklad pro účtování hovorů a pro zjišťování událostí v průběhu hovoru.

4.1 Call Detail Records (CDR)

CDR jsou podrobné záznamy, které zaznamenávají historii uskutečněných telefonních hovorů. Tyto záznamy lze využít různými způsoby. Jejich hlavní předností je, že poskytují podklady pro účtování hovorů, podklady pro analýzu objemu uskutečněných telefonních hovorů za časovou jednotku a mnohé další statistiky. CDR záznamy lze v Asterisku ukládat do souboru, databáze, atd. (více v kapitole 4.1.2). Základní konfiguraci CDR záznamů lze provádět v souboru `cdr.conf` .[2]

4.1.1 Formát CDR záznamů

CDR záznamy mají ve výchozím nastavení řadu atributů:

- **accountcode** - ID účtu
- **src** - ID volajícího
- **dst** - volané číslo
- **dcontext** - kontext volaného
- **clid** - ID volajícího včetně jeho jména
- **channel** - kanál volajícího
- **dstchannel** - kanál volaného
- **lastapp** - poslední provedená aplikace v dialplanu
- **lastdata** - data, která byla předána poslední provedené aplikaci
- **start** - čas zahájení hovoru
- **answer** - čas odpovědi na hovor
- **end** - čas ukončení hovoru
- **duration** - čas mezi začátkem a koncem hovoru
- **billsec** - čas mezi odpovědí na hovor a koncem hovoru (čas pro účtování hovorného)
- **disposition** - odpověď na výzvu k hovoru může být ve tvaru NO ANSWER, FAILED, BUSY, ANSWERED, nebo UNKNOWN

- **amaflags** - příznaky pro Automatic Message Accounting (AMA)
- **userfield** - uživatelsky definované pole
- **uniqueid** - unikátní ID pro kanál volajícího

4.1.2 Možnosti ukládání CDR záznamů do DBMS

Asterisk nabízí několik možností jak ukládat CDR záznamy, a také k tomu poskytuje nástroje. Jednou z možností je ukládání CDR záznamů do CSV souborů, které lze dále zpracovávat v tabulkových procesorech. Z důvodu mnohem efektivnější práce s daty je ukládání do databáze mnohem lepší variantou. Pro tuto možnost ukládání CDR záznamů Asterisk poskytuje následující moduly:

- **cdr_mysql** - pro MySQL
- **cdr_pgsql** - pro PostgreSQL
- **cdr_sqlite** - pro SQLite verze 2
- **cdr_sqlite3_custom** - pro SQLite verze 3
- **cdr_tds** - pro Sybase a MSSQL
- **cdr_odbc** - ukládání CDR přes ODBC, což je univerzální rozhraní k databázím
- **cdr_adaptive_odbc** - ukládání CDR přes ODBC rozhraní (lze měnit strukturu tabulky)

4.1.3 Ukládání CDR záznamů do PostgreSQL

Jednou z možností jak ukládat CDR záznamy, která byla zmíněna v kapitole 4.1.2, je využití databáze PostgreSQL. Buď bude databáze umístěna na stejném serveru jako Asterisk, nebo bude provozována na odděleném serveru. Při větších objemech hovorů se jeví druhá varianta jako lepší z důvodu velkého zatížení serveru. Nicméně nastíním první variantu, která je pro většinu PBX dostatečná. V první řadě je potřeba mít na serveru zprovozněnou databázi PostgreSQL. Instalace a důležité příkazy pro práci s databází PostgreSQL jsou zmíněny v příloze A.

Nejdříve se provede nová kompilace s podporou PostgreSQL:

```
$ ./configure --with-postgres=/usr
```

Po zkompilování zadáme příkaz:

```
$ make menuconfig
```

Nyní z menu vybereme možnost Call Detail Recording, dále zvolíme modul `cdr_pgsql` a poté menu opustíme (změny je nutné uložit) a provedeme instalaci:

```
$ make
$ make install
```

Tímto je modul `cdr_pgsql` nainstalován. Pokud není v `modules.conf` nastaveno níže uvedené, tak je potřeba doplnit `load=>cdr_pgsql.so`, čím zajistíme načtení tohoto modulu.

```
[modules]
autoload=yes
```

Dále je potřeba nastavit v `cdr_pgsql.conf`, kde se mají CDR záznamy ukládat:

```
[global]
hostname=localhost
port=5432
dbname=asterisk
password=password
user=asterisk
table=cdr
encoding=UTF8
timezone=UTC
```

Po reloadu Asterisku je možné ověřit, zda jsou CDR záznamy aktivní příkazem z CLI modu:

```
*CLI> cdr show status

Call Detail Record (CDR) settings
----------------------------------
  Logging:                    Enabled
  Mode:                       Simple
  Log unanswered calls:       No

* Registered Backends
  --------------------
    cdr-custom
    pgsql
```

V poslední řadě je potřeba vytvořit databázovou tabulku, do které se budou CDR záznamy načítat (zdrojový kód je uveden v příloze B.1). Konfigurace je hotova a při každém hovoru by mělo proběhnout jeho zaznamenání do databáze.

4.2 Channel Event Logging (CEL)

Channel event logging neboli CEL je nový systém, jehož cílem je zaznamenat nejen hovor, ale i celý jeho průběh. CEL je daleko pružnější systém než samotné CDR záznamy, jelikož dokáže zaznamenávat složitý průběh celého hovoru a poskytuje skutečný obraz událostí během hovoru. Pro CEL, stejně jako pro CDR záznamy Asterisk nabízí rozličné možnosti ukládání dat.[2]

Typ události	Popis
CHAN_START	vytvoření kanálu
CHAN_END	odstranění kanálu
LINKEDID_END	poslední kanál s daným ID byl odstraněn
ANSWER	odpověď na vytvoření kanálu (vyzvednutí hovoru)
HANGUP	zavěšení hovoru (krátce potom bude následovat CHAN_END)
APP_START	aplikace začala využívat kanál
APP_END	aplikace ukončila využívat kanál
PARK_START	hovor byl zaparkován
PARK_END	bylo opuštěno parkoviště
BRIDGE_START	přepojení hovoru a to v případě použití Dial() nebo Queue()
BRIDGE_END	přepojení bylo ukončeno
BRIDGE_UPDATE	změna údajů během přepojení
BLINDTRANSFER	byl proveden špatný transfer
ATTENDEDTRANSFER	byl proveden transfer
USER_DEFINED	tyto události jsou generovaný pomocí CELGenUserEvent(eventname)

Tabulka 2: Typy událostí v CEL

4.2.1 Formát CEL

Taktéž CEL obsahují spoustu atributů, zde je uveden jejich přehled:

- **eventtype** - typ události, popsáno v Tabulce 2
- **eventtime** - čas události
- **cid_name** - jméno volajícího spojené s událostí
- **cid_num** - číslo volajícího spojené s událostí
- **cid_ani** - automatické identifikační číslo (ANI) hovoru spojené s událostí
- **cid_rdnis** - přesměrovací číslo spojené s událostí
- **cid_dnid** - volané číslo spojené s událostí
- **exten** - směrování v dialplanu, které se právě provádí
- **context** - kontext v dialplanu, který se právě provádí
- **channame** - název kanálu spojeného s událostí
- **appname** - název aplikace z dialplanu, která se právě provádí
- **appdata** - předaná data, která byla předána poslední prováděné aplikaci

- **amaflags** - příznaky pro Automatic Message Accounting (AMA)
- **accountcode** - ID účtu
- **uniqueid** - unikátní ID pro kanál volajícího
- **userfield** - uživatelsky definované pole
- **linkedid** - toto ID pomáhá provázat více událostí, které jsou součástí jednoho hovoru
- **peer** - název kanálu propojeného s kanálem uvedeným v channame

4.2.2 Možnosti ukládání CEL do DBMS

Obdobně jako pro CDR záznamy, tak i pro CEL, Asterisk poskytuje moduly pro ukládání do databáze, zde je jejich přehled:

- **cel_mysql** - pro MySQL
- **cel_pgsql** - pro PostgreSQL
- **cel_sqlite** - pro SQLite verze 2
- **cel_sqlite3_custom** - pro SQLite verze 3
- **cel_tds** - pro Sybase a MSSQL
- **cel_odbc** - ukládání CEL přes ODBC, což je univerzální rozhraní k databázím
- **cel_adaptive_odbc** - ukládání CEL přes ODBC rozhraní (lze měnit strukturu tabulky)

4.2.3 Ukládání CEL do PostgreSQL

V této části, obdobně jako u CDR záznamů, nastíním základní postup pro zprovoznění ukládání CEL do databáze. Instalace probíhá obdobně jako v kapitole 4.1.3 pouze s tím rozdílem, že v menu vybereme možnost Channel Event Logging a dále modul `cdr_pqsql`. Po dokončení instalace je potřeba CEL aktivovat v `cel.conf`:

```
[general]
enable=yes
```

Dále je nutné provést následující konfiguraci v souboru `cel_pgsql.conf`:

```
[global]
hostname=localhost
port=5432
dbname=asterisk
password=password
user=asterisk
table=cel
```

Zda je CEL aktivní je možno ověřit v režimu CLI příkazem:

```
*CLI> cel show status
```

Instalace je hotova, nyní je pouze potřeba vytvořit databázovou tabulku CEL, zdrojový kód je uveden v příloze B.2.

5 Asterisk Manager Interface (AMI)

Asterisk Manager Interface (AMI) poskytuje rozhraní pro monitorování a řízení Asterisku. AMI umožňuje sledovat aktuální události (manager events) a také komunikovat s Asteriskem (manager actions). Funguje na modelu klient/server a základní nastavení je možné provádět v souboru `manager.conf`.[2]

5.1 Typy zpráv

V AMI jsou definovány dva základní druhy zpráv:

- Akce (manager actions)
- Události (manager events)

5.1.1 Akce

Akce jsou požadavky od klienta, při kterých je od Asterisku očekávána určitá činnost a následné odeslání informací s tím souvisejících. Například můžeme po Asterisku chtít, aby ukončil aktivní hovor nebo pomocí AMI provést příkaz v příkazové řádce (CLI) Asterisku.[2]

Syntaxe příkazů vypadá následovně:

```
Action: <typ akce><CRLF>
<parametr 1>: <hodnota 1><CRLF>
<parametr 2>: <hodnota 2><CRLF>
<CRLF>
```

Seznam dostupných akcí i s popisem je uveden v [23] nebo jej můžeme získat pomocí příkazu z příkazové řádky Asterisku:

```
*CLI> manager show commands
```

5.1.2 Události

Jedná se o jednosměrné zprávy vyvolané samotným Asteriskem, které směřují od ústředny ke klientovi. Události informují klienta o stavu systému.[2]

Například událost o ukončení hovoru může vypadat takto:

```
Event: Hangup
Privilege: call,all
Channel: SIP/60-00000003
Uniqueid: 1365062215.3
CallerIDNum: 60
CallerIDName: 60
ConnectedLineNum: <unknown>
ConnectedLineName: <unknown>
Cause: 0
Cause-txt: Unknown
```

5.2 Připojení k AMI

Existuje několik způsobů, jak se k AMI připojit. K nejčastějším patří připojení prostřednictvím TCP. Zda se připojení povedlo, bude demonstrováno pomocí telnetu. Nejdříve je však potřeba zprovoznit samotné AMI. Provedeme jen nejzákladnější a pouze ilustrační konfiguraci v souboru `/etc/asterisk/manager.conf`:

```
[general]
enabled = yes
;povoleni AMI z localhosta na portu 5038
port = 5038
bindaddr = 127.0.0.1

[modrypomeranc] ; vytvoreni uzivatelskeho uctu
secret = cernejablko
```

Nyní otestujeme připojení k AMI na portu 5038. Přihlásíme se pomocí akce Login, provedeme akci Ping a odhlásíme se akcí Logoff:

```
$ telnet localhost 5038
Trying 127.0.0.1...
Connected to localhost.
Escape character is '^]'.
Asterisk Call Manager/1.1
Action: Login
Username: modrypomeranc
Secret: cernejablko

Response: Success
Message: Authentication accepted

Event: FullyBooted
Privilege: system,all
Status: Fully Booted

Action: Ping

Response: Success
Ping: Pong
Timestamp: 1364981371.697166

Action: Logoff

Response: Goodbye
Message: Thanks for all the fish.
```

5.3 Ukončení aktivního hovoru

V případě napadení ústředny je dobré vědět jak ukončovat hovory[3] dříve než dojde k nějaké větší finanční škodě. V okamžiku zahájení hovoru u napadeného účtu se v CEL objeví záznam, kde je uveden typ události (eventtype) `CHAN_START`. Následující příklad demonstruje zahájení hovoru ze SIP účtu 60 na telefonní číslo 688811445.

```
  eventtype    | cid_name |   exten    |     channame
---------------+----------+------------+------------------
 CHAN_START    | 60       | 688811445 | SIP/60-00000002
```

V okamžiku vložení záznamu lze v AMI hovor ihned ukončit. Hovor se ukončuje pomocí akce `Hangup` a názvu kanálu.

```
ACTION: Hangup
Channel: SIP/60-00000002

Response: Success
Message: Channel Hungup
```

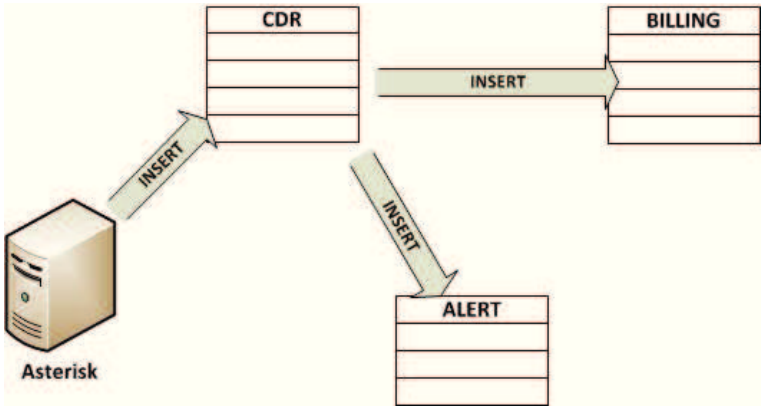
Následně po zadání akce `Hangup` dojde k okamžitému ukončení hovoru.

```
Event: Hangup
Privilege: call,all
Channel: SIP/61-00000003
Uniqueid: 1365967937.3
CallerIDNum: 688811445
CallerIDName: <unknown>
ConnectedLineNum: 60
ConnectedLineName: 60
Cause: 16
Cause-txt: Normal Clearing

Event: Dial
Privilege: call,all
SubEvent: End
Channel: SIP/60-00000002
UniqueID: 1365967937.2
DialStatus: CANCEL

Event: Hangup
Privilege: call,all
Channel: SIP/60-00000002
Uniqueid: 1365967937.2
CallerIDNum: 60
CallerIDName: 60
ConnectedLineNum: <unknown>
ConnectedLineName: <unknown>
Cause: 0
Cause-txt: Unknown
```

[3] od verze Asterisk 11.x je možné pomocí regulárního výrazu ukončit více hovorů zároveň

6 Návrh a implementace softwaru pro detekci zneužití ústředny

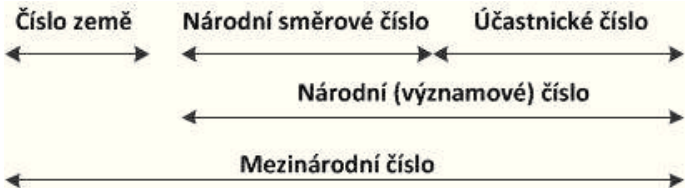
Systém se bude skládat ze dvou částí, a to tarifikaci a detekci zneužití. Tarifikace nám umožní identifikovat hovory, které budou vkládány ve formě CDR záznamů. Identifikaci hovoru nám umožní trigger nad tabulkou `CDR`, který bude spouštěn pro každý nový hovor. Tento trigger bude vkládat již identifikovaný hovor do tabulky `BILLING`. Před samotným vložením hovoru do tabulky `BILLING` bude dalším triggerem dopočtena jeho cena. U již identifikovaného hovoru uvidíme, z kterého účtu bylo voláno, kam bylo voláno, do jaké země bylo voláno a jaká byla cena hovoru. Systém pro detekci zneužití bude generovat alarmy ve formě záznamů do tabulky `ALERT`. Alarmy budou vkládány pomocí triggeru nad tabulkou `CDR`, který bude při každém dalším hovoru podle parametrů a statistik kontrolovat, zda již na daném účtě nebyly naplněny podmínky zneužití. Statistiky bude trigger získávat výpočtem z identifikovaných a vyčíslených hovorů, které budou uloženy v tabulce `BILLING`.



Obrázek 6: Struktura systému

6.1 Struktura telefonního čísla

Před samotným návrhem softwaru je nutné znát strukturu telefonního čísla (viz. Obrázek 7), podle něhož budou hovory rozlišeny na národní a mezinárodní, a následně stanovena cena hovoru.



Obrázek 7: Struktura telefonního čísla

- **Mezinárodní číslo** - Může dosáhnout délky 12 až 15 číslic. V České republice, až na výjimky, dosahuje maximální délky 12 číslic.
- **Číslo země** - Může mít 1 až 3 číslice. České republice bylo Mezinárodní telekomunikační unií (ITU) přiděleno číslo `420`. Čísla ostatních zemí jsou uvedena v literatuře [25].
- **Národní směrové číslo** - Je buď typu národního směrového čísla cílové sítě nebo národního směrového čísla telefonního obvodu.
- **Národní (významové číslo)** - Toto číslo se skládá z národního směrového čísla a účastnického čísla.
- **Mezinárodní přestupný znak** - Jedná se o znak, který je nutný předřadit před mezinárodní číslo. V České republice je tímto přestupným znakem `00`, někdy může být znakem + (plus).

Více je uvedeno v literatuře [24].

6.1.1 Mezinárodní hovor

Mezinárodní hovor od národního můžeme rozlišit podle mezinárodního rozlišovacího znaku. Pokud je zadán, až na výjimku se jedná o mezinárodní hovor. Za tuto výjimku lze považovat vytočení čísla s mezinárodním přestupným znakem a mezinárodní předvolbou stejného státu, ve kterém se právě nacházíme. Pro náš případ je to číslo `00420` nebo `+420` (ČR), tedy místní hovor.

6.1.2 Místní telefonní hovor

Místní hovor je takový, při kterém je vytočeno číslo bez mezinárodního přestupného znaku, tzn. začíná národním směrovým číslem. Druhá možnost je vytočení mezinárodního čísla, včetně přestupného znaku, ale číslo země bude `420` (tedy ČR).

6.2 Tarifikační systém

Důležitou součástí softwaru pro detekci zneužití ústředny je tarifikační systém. Tarifikační systém lze vytvořit na základě CDR nebo CEL. V této části bude popsáno vytvoření softwaru pomocí CDR záznamů s využitím databáze PostgreSQL (základní příkazy jsou uvedeny v příloze A) a procedurálního jazyka PL/pgSQL (úvod do jazyka PL/pgSQL je popsán v [19]).

6.2.1 Tabulka "PREFIX_COUNTRY"

Nejdříve je potřeba vytvořit tabulku, podle které budou účtovány mezinárodní hovory. Sloupce v tabulce představují název země, číslo země, účtování první tarifikační periody, účtování dalších tarifikačních period a cenu za minutu. Zdrojový kód a příklad výsledné tabulky s daty bude vypadat následovně:

```
CREATE TABLE PREFIX_COUNTRY(
COUNTRY VARCHAR(10) NOT NULL,
PREFIX VARCHAR(5) NOT NULL,
TARIFFICATION_1 NUMERIC NOT NULL,
TARIFFICATION_2 NUMERIC NOT NULL,
COST_MINUTE NUMERIC NOT NULL);
```

Výpis 1: Tabulka prefix_country

country	prefix	tariffication_1	tariffication_2	cost_minute
SK	421	60	1	0.70
DE	49	1	1	0.70
CN	86	60	60	1.10
QA	974	60	1	21

6.2.2 Tabulka "PREFIX_LOCAL"

Pomocí další tabulky nejdříve rozlišíme místní hovor podle národního směrového čísla na jednotlivé telefonní sítě. Sloupce v tabulce představují národní směrové číslo a název operátora. Zdrojový kód a příklad výsledné tabulky s daty bude vypadat následovně:

```
CREATE TABLE PREFIX_LOCAL(
PREFIX VARCHAR(5) NOT NULL,
OPERATOR VARCHAR(2) NOT NULL);
```

Výpis 2: Tabulka prefix_local

prefix	fk_operator
800	ZL
601	O2
730	TM
777	VF

6.2.3 Tabulka "PREFIX_OPERATOR"

Další tabulka, podle rozlišení na jednotlivé operátory, určuje cenu místního hovoru. Její sloupce představují název operátora, účtování první tarifikační periody, účtování dalších tarifikačních period a cenu za minutu. Zdrojový kód a příklad výsledné tabulky s daty bude vypadat následovně:

```
CREATE TABLE PREFIX_OPERATOR(
OPERATOR VARCHAR(2) NOT NULL,
TARIFFICATION_1 NUMERIC NOT NULL,
TARIFFICATION_2 NUMERIC NOT NULL,
COST_MINUTE NUMERIC NOT NULL);
```

Výpis 3: Tabulka prefix_operator

```
 operator | tariffication_1 | tariffication_2 | cost_minute
----------+-----------------+-----------------+-------------
 ZL       |               1 |               1 |           0
 PL       |               1 |               1 |        0.65
 O2       |              60 |               1 |        2.95
 TM       |              60 |               1 |        2.95
 VF       |              60 |               1 |        2.95
```

6.2.4 Tabulka "BILLING"

Poslední tabulka již bude zaznamenávat ceny za jednotlivé hovory. Tuto tabulku budou plnit triggery uvedené v kapitolách 6.2.5 a 6.2.6. Jednotlivé sloupce tabulky představují časový údaj, číslo volajícího, číslo volaného, účtovaný čas, název země, název operátora a účtovanou cenu za hovor. Zdrojový kód bude vypadat následovně:

```
CREATE TABLE BILLING(
DATE timestamp NOT NULL,
SOURCE VARCHAR(20) NOT NULL,
DESTINATION VARCHAR(20) NOT NULL,
BILLED_TIME INT NOT NULL,
COUNTRY_PREFIX VARCHAR(3) NULL,
OPERATOR_PREFIX VARCHAR(3) NULL,
COST NUMERIC NOT NULL);
```

Výpis 4: Tabulka prefix_billing

6.2.5 Trigger "T_BILLING"

Dále je nutné vytvořit trigger nad tabulkou `CDR`, který využije data vložená do tabulky `CDR` a po jejich úpravě a dopočtu dalších dat je vloží do tabulky `BILLING`. Nejdříve je potřeba sjednotit mezinárodní přístupný znak, a poté můžeme rozlišit mezinárodní hovor od místního, což bylo popsáno v kapitolách 6.1.1 a 6.1.2.

```
IF (substr(V_POMCHAR,1,3)='420') THEN V_DESTINATION:=ltrim(V_POMCHAR,'00420');
END IF;

IF (substr(V_DESTINATION,1,2))='00'
   --international call
   .
   .
ELSE
   --local call
   .
   .
END IF;
```

Výpis 5: Trigger t_billing - odlišení hovorů

V případě, že se jednalo o mezinárodní hovor, je potřeba přidat k hovoru název země, do které bylo voláno. To to provedeme pomocí selectu z tabulky "PREFIX_COUNTRY", kde přiřazení proběhne podle mezinárodního směrového čísla. Pokud se jednalo o místní hovor, název země byl automaticky nastaven na "CZ" a provede se select z tabulky "PREFIX_LOCAL", který podle národního směrového čísla přiřadí k hovoru název operátora.

6.2.6 Trigger "T_BILLING_COST"

Nyní jsou v tabulce BILLING všechny údaje, až na cenu za hovor. Doplnění ceny za hovor má právě za úkol trigger T_BILLING_COST. Jedná se trigger nad tabulkou BILLING, jež na základě zjištěných údajů, které do ní budou vloženy, spočítá cenu za uskutečněný hovor.

```
IF (V_TIME_SEC > 0) THEN
--first zone of tarification
V_COST_UNIT:=(V_COST_MINUTE/(60/V_TARIFFICATION_1));
new.COST:=V_COST_UNIT;
ELSE
new.COST:=0;
END IF;
IF (V_TIME_SEC > V_TARIFFICATION_1) THEN
--second zone of tarification
   V_COST_UNIT:=(V_COST_MINUTE/(60/V_TARIFFICATION_2));
   V_TIME_SEC:=V_TIME_SEC-V_TARIFFICATION_1;
   new.COST:=new.COST + (ceil(V_TIME_SEC/V_TARIFFICATION_2)*V_COST_UNIT);
END IF;
new.COST:=round(new.COST,2);
```

Výpis 6: Trigger t_billing_cost

Vzorec pro výpočet ceny hovoru:

– hovor kratší než je délka první tarifikační periody:

$$C_1 = \frac{C_m}{\frac{60}{D_1}}$$

$$C = C_1$$

– hovor delší než je délka první tarifikační periody:

$$C_1 = \frac{C_m}{\frac{60}{D_1}}$$

$$C_2 = \frac{C_m}{\frac{60}{D_2}}$$

$$T_2 = T - D_1$$

$$\mathrm{C} = C_1 + \left(\frac{T_2}{D_2}\right) * C_2$$

$\mathbf{C}$– cena hovoru

$\mathbf{C_1}$– cena za první tarifikační periodu

$\mathbf{C_2}$– cena za další tarifikační periodu

$\mathbf{C_m}$– cena za minutu

$\mathbf{D_1}$– délka první tarifikační periody

$\mathbf{D_2}$– délka dalších tarifikačních period

$\mathbf{T}$– délka hovoru

$\mathbf{T_2}$– provolaný čas v dalších tarifikačních periodách

Nejdříve je nutné provést dva různé selecty pro zjištění tarifikace a ceny za minutu. Jeden je pro mezinárodní hovor a druhý pro místní. Cena hovoru je dána součtem cen za jeho první tarifikační periodu a další periody. V praxi se většinou používají periody od 1 do 120 sekund, přičemž nejpoužívanější tarifikace jsou 1+1, 60+1, 60+30, 60+60, 120+1, 120+60 a 120+120 sekund. Pro každou tarifikační periodu se bude cena počítat zvlášť. Pokud délka hovoru nepřesáhne délku první tarifikační periody, počítá se pouze cena za první periodu. U hovorů delších než první tarifikační perioda se nejdříve spočítá cena za první periodu, tedy obdobně jako u hovorů kratších než první tarifikační perioda, a poté se přičte cena za zbytek provolaného času účtovaného podle délky dalších period. Například u hovoru o délce 1 minuta a 35 sekund při tarifikaci 60+30 se v rámci první tarifikační periody bude účtovat 60 sekund a zbývajících 35 sekund v rámci dalších period, tedy dvakrát 30 sekund a celkový účtovaný čas je roven 2 minutám.

Po vyplnění všech údajů, by tabulka `BILLING` mohla vypadat následovně:

```
        date         | source |  destination   | billed_time
---------------------+--------+----------------+-------------
 2013-04-04 16:35:56 | 60     | 608811536      |          11
 2013-04-04 16:36:20 | 60     | 608811596      |          70
 2013-04-05 08:02:28 | 60     | 00493514820980 |          62
 2013-04-05 08:05:24 | 60     | 00421608811776 |          68
 country_prefix | operator_prefix | cost
----------------+-----------------+------
 CZ             | VF              | 2.95
 CZ             | VF              | 3.44
 DE             |                 | 0.72
 SK             |                 | 0.79
(4 rows)
```

6.3 Detekce zneužití

Další částí systému je samotná detekce zneužití. Tato část bude generovat alarmy, které upozorní na potenciální možnost zneužití daného účtu. Systém lze vytvořit taktéž na základě CDR nebo CEL. V této části bude popsáno vytvoření softwaru pomocí CDR záznamů a identifikovaných hovorů z tabulky `BILLING` taktéž s využitím databáze PostgreSQL a procedurálního jazyka PL/pgSQL.

6.3.1 Tabulka "ALERT"

Nejdříve je nutné vytvořit databázovou tabulku, ve které budou evidovány varování o tom, že ústředna může být napadena útočníkem. Jednotlivé sloupce tabulky představují datum vzniku varování, název účtu, typ varování a popis toho, co bylo detekováno.

```
CREATE TABLE ALERT(
DATE timestamp not null,
ACCOUNT VARCHAR(40) not null,
TYPE VARCHAR(20) not null,
NOTICE VARCHAR(50) null);
```

Výpis 7: Tabulka alert

6.3.2 Trigger "T_ALERT"

Nyní je nutné vytvořit trigger nad tabulkou `CDR`, který bude analyzovat záznamy uskutečněných hovorů. Trigger sleduje následující parametry:

- Maximální provolaná částka
- Vysoký poměr hovorů do drahých lokalit
- Zvýšení objemu hovorů

V případě překročení limitů u těchto parametrů trigger vloží záznam do tabulky `ALERT`, ale nejvýše jedenkrát za 30 minut. V tabulce bude podle typu rozlišeno, jestli se jedná pouze o varování (bude uveden typ `WARNING`) nebo již o vážné nebezpečí (bude uveden typ `DANGER`).

Maximální provolaná částka

Mezi základní opatření, která chrání před zneužitím účtu i před nevědomostí klienta, patří omezení maximální provolané částky na jednoho klienta za časovou jednotku. V našem případě to bude měsíc, takže každý průchod triggeru zjistí u klienta aktuální provolanou částku v daném měsíci. Pokud dosáhne 75% limitní částky, která byla nastavena na 10000, bude spuštěn alarm typu `WARNING`. Pokud již dosáhne 100%, bude spuštěn alarm typu `DANGER`.

```
SELECT sum(cost) INTO V_SUM_COST FROM billing WHERE source=new.src
   AND date_trunc('month', date)=date_trunc('month', new.calldate);
IF ( V_SUM_COST > (V_MAX_BILLING*0.75) AND V_SUM_COST < V_MAX_BILLING) THEN
   IF NOT EXISTS (SELECT * FROM alert WHERE account=new.src
   AND date BETWEEN (new.calldate - interval '30 minutes') AND new.calldate
   AND NOTICE ='exhausted 75% of the billing limit' ) THEN
      INSERT INTO Alert(date,account,type,notice)
      VALUES(new.calldate,new.src,'WARNING','exhausted 75% of the billing limit');
   END IF;
ELSIF (V_SUM_COST > V_MAX_BILLING) THEN
   IF NOT EXISTS (SELECT * FROM alert WHERE account=new.src
   AND date BETWEEN (new.calldate - interval '30 minutes') AND new.calldate
   AND NOTICE ='exhausted 100% of the billing limit') THEN
      INSERT INTO Alert(date,account,type,notice)
      VALUES(new.calldate,new.src,'DANGER','exhausted 100% of the billing limit');
   END IF;
END IF;
```

Výpis 8: Trigger t_alert - maximální provolaná částka

Vysoký poměr hovorů do drahých lokalit

Dalším opatřením je sledování počtu hovorů do drahých zahraničních destinací. Nejdříve zjistíme množství všech provolaných sekund za 24 hodin pro daného klienta, a pak počet provolaných sekund do drahých zahraničních destinací za uplynulých 24 hodin. Nyní je podělíme, vynásobíme 100 a dostaneme poměr hovorů do drahých zahraničních lokalit vyjádřený v procentech, přičemž je nastaven minimální validní vzorek provolaných sekund. Pokud poměr hovorů do drahých lokalit přesáhne 20%, je spuštěn alarm typu `WARNING`, jestliže již přesáhne 30%, je spuštěn alarm typu `DANGER`.

```
SELECT sum(billed_time) INTO V_SUM_CALLS FROM billing WHERE source=new.src
   AND date >= (new.calldate - interval '24 hours') ;
IF (V_SUM_CALLS > V_MIN_SUM_CALLS) THEN
   SELECT sum(billed_time) INTO V_SUM_EXP_CALLS FROM billing WHERE source=new.src
      AND date >= (new.calldate - interval '24 hours')
      AND country_prefix IN (SELECT country from prefix_country WHERE cost_minute > 15);
   V_PER_EXP_CALLS:=round(((V_SUM_EXP_CALLS/V_SUM_CALLS)*100),0);
   IF (V_PER_EXP_CALLS > 20 AND V_PER_EXP_CALLS < 30 ) THEN
      IF NOT EXISTS (SELECT * FROM alert WHERE account=new.src
      AND date BETWEEN (new.calldate - interval '30 minutes') AND new.calldate
      AND NOTICE ='over 20% call to expensive locations') THEN
         INSERT INTO Alert(date,account,type,notice)
         VALUES(new.calldate,new.src,'WARNING',' over 20% call to expensive locations');
      END IF;
   ELSIF (V_PER_EXP_CALLS > 30) THEN
      IF NOT EXISTS (SELECT * FROM alert WHERE account=new.src
      AND date BETWEEN (new.calldate - interval '30 minutes') AND new.calldate
      AND NOTICE ='over 30% call to expensive locations') THEN
         INSERT INTO Alert(date,account,type,notice)
         VALUES(new.calldate,new.src,'DANGER','over 30% call to expensive locations');
```

```
            END IF;
        END IF;
    END IF;
```

Výpis 9: Trigger t_alert - vysoký poměr hovorů do drahých lokalit

Zvýšení objemu hovorů

Jako poslední můžeme přidat opatření, které bude sledovat nárůst objemu hovorů. Nejdříve zjistíme množství provolaných sekund za posledních 24 hodin daným klientem a vypočteme hodinový průměr. Dále si zjistíme množství provolaných sekund za poslední hodinu. Pokud počet provolaných sekund v dané hodině byl větší než minimum, které zajišťuje validitu údajů, spočteme hodinový přírůstek hovorů. V případě přírůstku většího než 200% bude spuštěn alarm typu `WARNING` a v okamžiku přesáhnutí 400% bude spuštěn alarm typu `DANGER`.

```
SELECT sum(billed_time) INTO V_SUM_CALLS_LAST_HOUR
    FROM billing WHERE source=new.src AND date >= (new.calldate - interval '1 hour');
  V_AVG_CALLS_PER_HOUR:=(V_SUM_CALLS/24);
  IF (V_AVG_CALLS_PER_HOUR > V_MIN_SUM_CALLS_PER_HOUR) THEN
    V_INCREASE:=round((((V_SUM_CALLS_LAST_HOUR/V_AVG_CALLS_PER_HOUR)*100)
        -100),0);
    IF (V_INCREASE > 100 AND V_INCREASE < 200) THEN
      IF NOT EXISTS (SELECT * FROM alert where account=new.src
      AND date BETWEEN (new.calldate - interval '30 minutes') AND new.calldate
      AND NOTICE ='the number of calls has increased by 200%') THEN
        INSERT INTO Alert(date,account,type,notice)
        VALUES(new.calldate,new.src,'WARNING','the number of calls has increased by
            200%');
      END IF;
    ELSIF (V_INCREASE > 200) THEN
      IF NOT EXISTS (SELECT * FROM alert where account=new.src
      AND date BETWEEN (new.calldate - interval '30 minutes') AND new.calldate
      AND NOTICE ='the number of calls has increased by 400%') THEN
        INSERT INTO Alert(date,account,type,notice)
        VALUES(new.calldate,new.src,'DANGER','the number of calls has increased by
            400%');
      END IF;
    END IF;
  END IF;
```

Výpis 10: Trigger t_alert - zvýšení objemů hovorů

Nyní máme systém na detekci útoků hotov. Pomocí alarmů, které nám tento systém generuje, můžeme v okamžiku zjištění napadení jednotlivých účtů zajistit, že každý další hovor bude ukončen (ukončení hovoru pomocí AMI je uvedeno v kapitole 5.3).

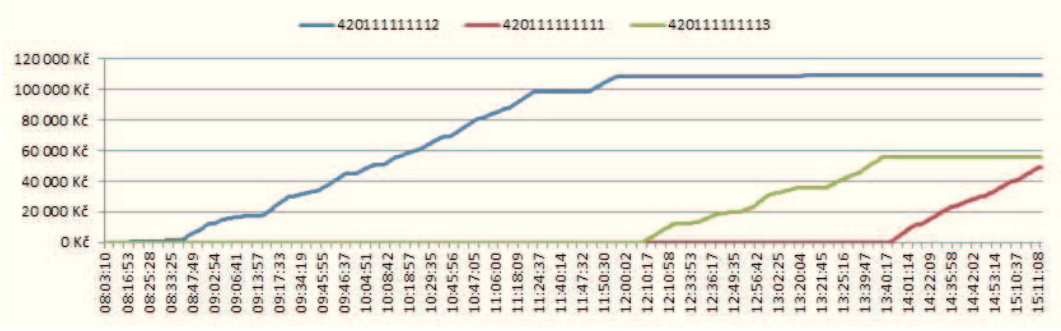
7 Testování softwaru pro detekci zneužití ústředny

V této kapitole bude popsáno testování softwaru na reálných útocích ve formě CDR záznamů, u nichž byla pozměněna pouze telefonní čísla. Testování bylo provedeno importem dat z CSV souboru do tabulky `CDR`. Nejdříve vždy proběhne krátká analýza detekce útoku. Za ní následuje výstup z tabulky `ALERT` a grafy, které znázorňují u jednotlivých účtů provolanou částku, poměr hovorů do drahých zahraničních lokalit a objem hovorů v časové ose.

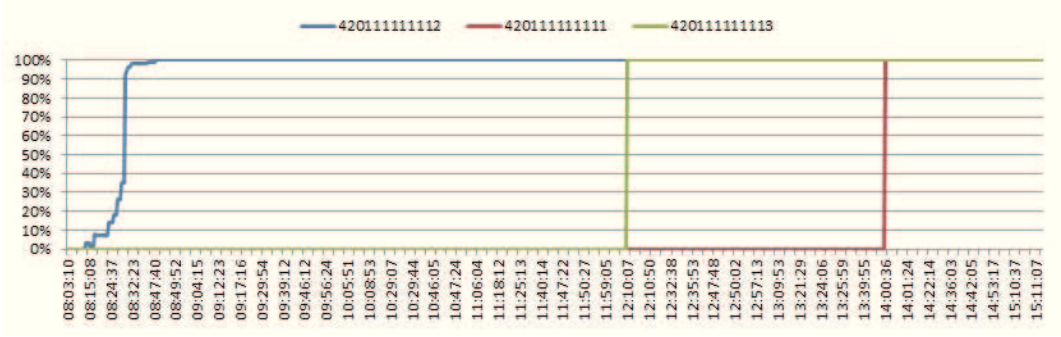
7.1 Útok č.1

Tento útok můžeme označit jako Toll Fraud, který se vyznačuje zneužitím účtů k volání do drahých zahraničních destinací. Zde se jedná zejména o hovory na telefonní čísla s číslem země 221 tedy Senegal. Zneužití bylo detekováno celkem na třech účtech. Jako první došlo k detekci zneužití na účtě 420111111112 v čase 08:45:57 vysokým poměrem hovorů do drahých zahraničních lokalit. K nárůstu těchto hovorů došlo již v čase kolem 08:30:00, ale objem uskutečněných hovorů byl velmi nízký. Dále byl na stejném účtě v čase 08:47:47 detekován prudký nárůst hovorů. Jako poslední byl v 08:48:03 spuštěn alarm z důsledku vyčerpání 75% volacího limitu a v čase 08:48:03 došlo k vyčerpání již 100%. Druhý účet, kde bylo detekováno zneužití, je 420111111113. Zde byl jako první v 12:10:17 spuštěn alarm, který detekoval vysokou míru hovorů do drahých zahraničních lokalit. Na stejném účtu byl detekován v čase 12:10:35 i velký nárůst hovorů a v 12:10:35 dosáhla provolaná částka již 100% volacího limitu. Posledním zneužitým účtem byl 420111111111. U tohoto účtu byla, obdobě jako u předcházejících dvou, nejdříve detekována vysoká míra hovorů do drahých zahraničních lokalit, a pak také vysoký nárůst objemu hovorů a vyčerpání maximálního volacího limitu.

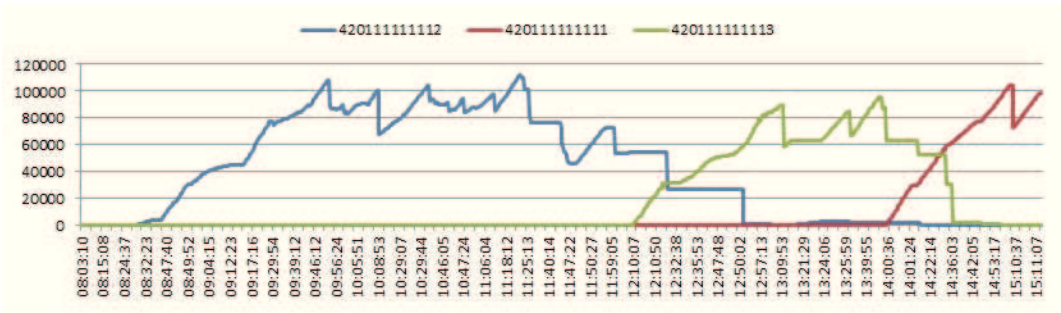
date	account	type	notice
2013-02-16 08:45:57	420111111112	DANGER	over 30% call to expensive locations
2013-02-16 08:47:47	420111111112	DANGER	the number of calls has increased by 400%
2013-02-16 08:48:03	420111111112	WARNING	exhausted 75% of the billing limit
2013-02-16 08:48:43	420111111112	DANGER	exhausted 100% of the billing limit
2013-02-16 12:10:17	420111111113	DANGER	over 30% call to expensive locations
2013-02-16 12:10:35	420111111113	DANGER	the number of calls has increased by 400%
2013-02-16 12:10:45	420111111113	WARNING	exhausted 75% of the billing limit
2013-02-16 12:10:58	420111111113	DANGER	exhausted 100% of the billing limit
2013-02-16 14:00:43	420111111111	DANGER	over 30% call to expensive locations
2013-02-16 14:01:05	420111111111	DANGER	the number of calls has increased by 400%
2013-02-16 14:01:14	420111111111	WARNING	exhausted 75% of the billing limit
2013-02-16 14:01:25	420111111111	DANGER	exhausted 100% of the billing limit



Obrázek 8: Útok č.1 - Provolaná částka



Obrázek 9: Útok č.1 - Poměr hovorů do drahých lokalit



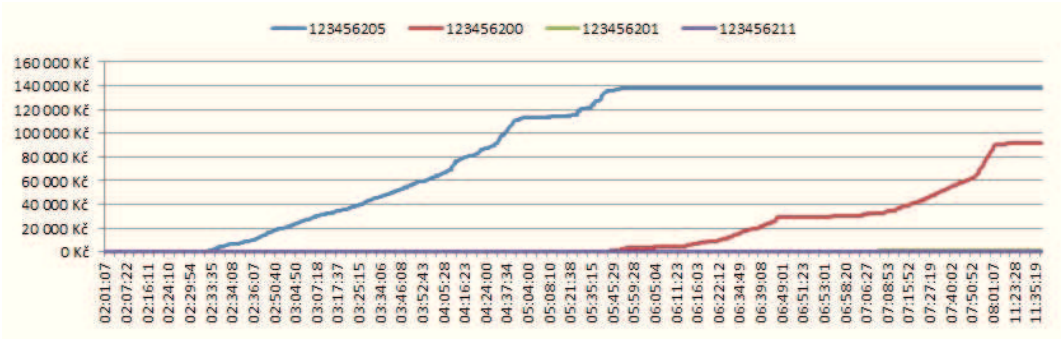
Obrázek 10: Útok č.1 - Objem hovorů za hodinu (v provolaných sekundách za hodinu)

7.2 Útok č.2

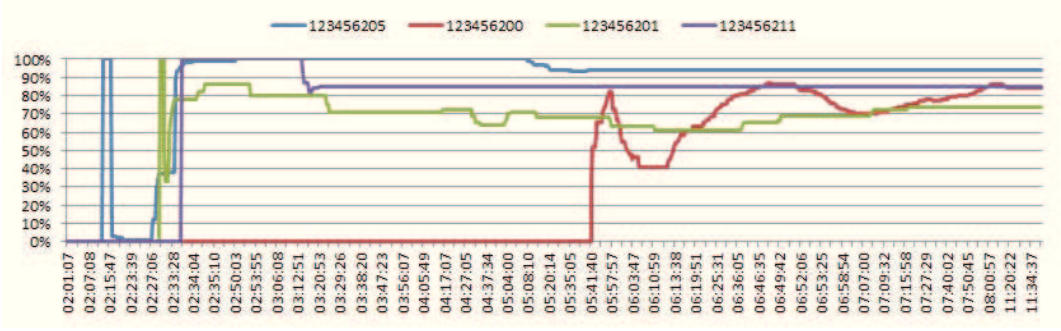
Jedná se taktéž o Toll Fraud. Hovory jsou nejčastěji směrovány do lokality s číslem země 221, což je Maroko a Západní Sahara. Zde došlo ke zneužití celkem dvou účtů. Jako první byl zneužit účet 123456205, u kterého byla v čase 02:33:36 detekována vysoká míra hovorů do drahých zahraničních lokalit. K překročení hranice 30% došlo již dříve, ale objem hovorů byl příliš nízký. V 02:34:04 byl u stejného účtu spuštěn alarm oznamující velký nárůst objemu hovorů. Později byly

spuštěny alarmy upozorňující na vyčerpání 75% a 100 % volacího limitu. U účtu 123456200 bylo jako první detekováno zneužití v čase 05:45:33, a to dosažením vysoké míry hovorů do drahých zahraničních lokalit. Dále byl v 06:00:13 detekován nárůst hovorů o více než 400%. Nakonec bylo v čase 06:16:41 detekováno vyčerpání 75% volacího limitu, který byl o 6 minut později zcela vyčerpán. U účtů 123456201 a 123456211 sice došlo k překročení 30% míry u hovorů do drahých zahraničních lokalit, ale jednalo se pouze o jednotky hovorů, tudíž nebyl spuštěn žádný alarm.

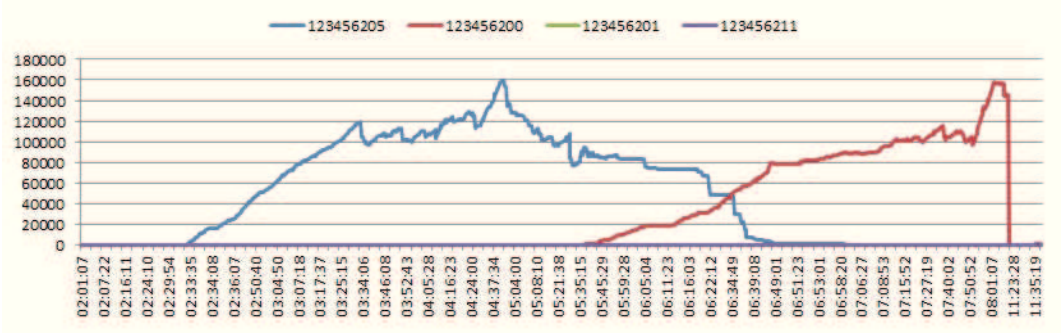
date	account	type	notice
2012-11-10 02:33:36	123456205	DANGER	over 30% call to expensive locations
2012-11-10 02:34:04	123456205	DANGER	the number of calls has increased by 400%
2012-11-10 02:34:59	123456205	WARNING	exhausted 75% of the billing limit
2012-11-10 02:36:46	123456205	DANGER	exhausted 100% of the billing limit
2012-11-10 05:45:33	123456200	DANGER	over 30% call to expensive locations
2012-11-10 06:00:13	123456200	DANGER	the number of calls has increased by 400%
2012-11-10 06:16:41	123456200	WARNING	exhausted 75% of the billing limit
2012-11-10 06:22:49	123456200	DANGER	exhausted 100% of the billing limit



Obrázek 11: Útok č.2 - Provolaná částka



Obrázek 12: Útok č.2 - Poměr hovorů do drahých lokalit



Obrázek 13: Útok č.2 - Objem hovorů za hodinu (v provolaných sekundách za hodinu)

Software pro detekci zneužití fungoval podle očekávání. Zneužití bylo detekováno v obou případech a u všech napadených účtů velmi časně, což by zásadně přispělo ke snížení způsobených finančních škod.

8 Závěr

V úvodu byl představen SW Asterisk, na který byla celá práce cílena. Byly předvedeny formy zneužití PBX a analyzovány dnes často opomíjené základní hrozby VoIP telefonie se zaměřením právě na tento software. Následně byly vyjmenovány způsoby jak se před těmito hrozbami účinně bránit. Dále byly v práci popsány možnosti logování hovorů do databáze, které jsou stěžejní pro detekci zneužití. Popis se týkal dvou subsystémů, které SW Asterisk standardně pro tuto činnost poskytuje, a to CDR (Call Detail Records) a CEL (Channel Event Logging). Byly ukázány základní funkce AMI (Asterisk Manager Interface), včetně možnosti ukončení aktivního hovoru, která se v případě zneužití PBX zdá jako velmi užitečná. Byl navržen software pro detekci zneužití PBX, jehož součástí byl tarifikační systém a systém na samotnou detekci zneužití. Tarifikační systém napomohl k identifikaci a vyčíslení ceny jednotlivých hovorů, které jsou evidovány v CDR záznamech. Systém pro samotnou detekci zneužití sledoval objemy a typy hovorů a v případě nestandardního chování na tuto skutečnost upozorňoval alarmy. V poslední části byl software otestován na reálných útocích. Pomocí představených součástí Asterisku bylo možné navrhnout a implementovat software, který umožní efektivní způsob detekce zneužití na SW PBX Asterisk a zamezení případným finančním škodám.

Software by bylo vhodné ještě otestovat v reálném provozu PBX, aby bylo zamezeno případným chybám v detekci zneužití. Dále by mohl být s využitím AMI (Asterisk Manager Interface) a CEL (Channel Event Logging) systém rozšířen o možnost ukončení už bežících drahých hovorů ve chvíli, kdy dojde k překročení limitu. Jako další možnost rozšíření se nabízí vytvoření grafického uživatelského rozhraní (GUI), kde by byla prezentována data ze systému pro detekci zneužití.

I přes mnohá úskalí, která jsou spojena s bezpečností VoIP si myslím, že má tato technologie velkou budoucnost, a to hlavně v business sféře. Zde může ušetřit nemalé peníze za nové ústředny, které se pro VoIP dají koupit mnohem levněji z důvodu možných open source řešení. Další přednosti jsou ceny hovorného a nové možnosti, které tato technologie poskytuje. S rozvojem pokrytí veřejných míst Wi-Fi a WiMax sítěmi a s větší dostupností VoIP služeb pro mobilní zařízení se dá očekávat, že mnoho VoIP hovorů bude uskutečněno právě přes tato zařízení. Zkrátka technologie VoIP má své místo v budoucnosti zajištěné.

Aleš Procházka

9 Reference

[1] DAVENPORT, Malcolm. *A Brief History of the Asterisk Project* [online]. c2010, poslední revize 23.10.2010 [cit.2013-03-05]. Dostupné z: <https://wiki.asterisk.org/wiki/display/AST/A+Brief+History+of+the+Asterisk+Project>.

[2] MADSEN, Leif; VAN MEGGELEN, Jim and BRYANT, Russell. *Asterisk™: The Definitive Guide*. Third Edition: O'Reilly Media, 2011. ISBN 978-0-596-51734-2.

[3] GONÇALVES, Flavio E. *Asterisk PBX Configuration Guide*: V.Office Networks Ltda. , 2010. ISBN 978-85-906904-2-9.

[4] *IVR | Asterisk.org* [online] [cit.2013-03-07]. Dostupné z: <http://www.asterisk.org/get-started/applications/ivr>.

[5] WIJA, Tomáš; ZUKAL, David and VOZŇÁK, Miroslav.*Asterisk Open source PBX* [online]. 2005 [cit.2013-03-10]. Dostupné z: <http://homel.vsb.cz/~voz29/files/voz_72.pdf>.

[6] PUŽMANOVÁ,Rita. *TCP-IP v kostce*: Kopp, 2004. 450 s. ISBN 80-7232-236-2.

[7] *SIP | CESNET* [online]. poslední revize 11.3.2012 [cit.2013-03-10]. Dostupné z: <https://sip.cesnet.cz/cs/protokoly/sip>.

[8] *Debian – Verze Debianu* [online]. c1997, poslední revize 23.5.2012 [cit.2013-03-06]. Dostupné z: <http://www.debian.org/releases/index.cs.html>.

[9] BRYANT, Russell. *Asterisk Versions* [online]. c2010, poslední revize 2.10.2012 [cit.2013-03-07]. Dostupné z: <https://wiki.asterisk.org/wiki/display/AST/Asterisk+Versions>.

[10] *Asterisk dimensioning - voip-info.org* [online]. poslední revize 28.6.2012 [cit.2013-03-07]. Dostupné z: http://www.voip-info.org/wiki/view/Asterisk+dimensioning.

[11] *Asterisk Downloads | Aterisk.org* [online] [cit.2013-03-07]. Dostupné z: <http://www.asterisk.org/downloads>.

[12] *Základní instalace Asterisk na Debian 6* (Squeeze) [online]. poslední revize 22.12.2010 [cit.2013-03-07]. Dostupné z: <http://wiki.4smart.cz/doku.php?id=asterisk-zakladni-instalace-asterisk-na-debian-6-squeeze>.

[13] Rouse, Margaret. *What is SPIT (spam over Internet telephony) ?* [online]. poslední revize 1.11.2008 [cit.2013-04-05]. Dostupné z: <http://searchunifiedcommunications.techtarget.com/definition/SPIT>.

[14] UNUTH, Nadeem. *VoIP Phishing - What is VoIP Phishing and How Does It Work* [online] [cit.2013-04-06]. Dostupné z: <http://voip.about.com/od/security/a/VoIPPhishing.htm>.

[15] *Rizika VoIP, bezpečnostní pravidla Asterisku* [online] [cit.2013-04-06]. Dostupné z: <http://www.volny.cz/bobanl/data/1101_01_AsterStupidAndSecur.pdf>.

[16] TONCAR, Vladimír. *Jak zabezpečit VoIP ústřednu před útokem?* [online]. poslední revize 15.11.2011 [cit.2013-04-07]. Dostupné z: <http://www.ictmanazer.cz/2011/11/jak-zabezpecit-voip-ustrednu-pred-utokem>.

[17] *Základní konfigurace Linux firewallu pomocí Iptables* [online]. poslední revize 10.10.2009 [cit.2013-04-08]. Dostupné z: <http://www.abclinuxu.cz/blog/Debian_Lenny/2009/10/zakladni-konfigurace-linux-firewallu-pomoci-iptables>.

[18] BOTOŠ, Csaba.*Vše o iptables* [online]. poslední revize 10.1.2006 [cit.2013-04-08]. Dostupné z: <http://www.root.cz/clanky/vse-o-iptables-uvod>.

[19] STĚHULE, Pavel. *Jemný úvod do jazyka PL/pgSQL PostgreSQL* [online] [cit.2013-03-22]. Dostupné z: <http://postgresql.ok.cz/doc/plpgsql.html>.

[20] *PostgreSQL 8.4.17 Documentation* [online] [cit.2013-03-22]. Dostupné z: <http://www.postgresql.org/docs/8.4/static/index.html>.

[21] DAVENPORT, Malcolm. *PostgreSQL CDR Backend* [online]. poslední revize 5.6.2012 [cit.2013-04-03]. Dostupné z: <https://wiki.asterisk.org/wiki/display/AST/PostgreSQL+CDR+Backend>.

[22] DAVENPORT, Malcolm. *PostgreSQL CEL Backend* [online]. poslední revize 31.8.2010 [cit.2013-04-03]. Dostupné z: <https://wiki.asterisk.org/wiki/display/AST/PostgreSQL+CEL+Backend>.

[23] *Asterisk manager API* [online]. poslední revize 31.11.2012 [cit.2013-03-20]. Dostupné z: <http://www.voip-info.org/wiki/view/Asterisk+manager+API>.

[24] *Číslovací plán veřejných telefonních sítí* [online]. poslední revize 25.9.2000 [cit.2013-04-10]. Dostupné z: <http://www.ctu.cz/1/download/cislovaci-plan-verejnych-telefonnich-siti_1114435245.pdf>.

[25] *List of country calling codes* [online].[cit.2013-04-11]. Wikipedie : otevřená encyklopedie. Dostupné z: <http://en.wikipedia.org/wiki/List_of_country_calling_codes>.

[26] *Asterisk PBX security risks and how to avoid being hacked - part 1* [online]. poslední revize 5.10.2010 [cit.2013-04-03]. Dostupné z: <http://kb.smartvox.co.uk/asterisk/secure-asterisk-pbx-part-1/>.

A Databáze PostgreSQL

Tato příloha popisuje základní instalaci a nejzákladnější příkazy, které jsou nezbytné pro zprovoznění logování hovorů do RDBMS PostgreSQL.

A.1 Instalace PostgreSQL

Nejdříve je potřeba nainstalovat následující balíky, které vytvoří databázový server:

```
$ apt-get install postgresql postgresql-client libpqxx-dev
```

Dále je nutné vytvořit databázi, ve které budeme pracovat, ale nejdříve musíme změnit uživatele na postgres:

```
$ su postgres
$ createdb asterisk  # vytvorime databazi asterisk
```

Nyní vytvoříme uživatele:

```
$ createuser -A -D -P asterisk  # vytvorime uzivatele asterisk
```

Abychom měli možnost používat procedurální jazyk PL/pgSQL, musí se pro danou databázi tento jazyk povolit, což umožňuje příkaz:

```
$ createlang plpgsql asterisk
```

Zda je jazyk pro danou databázi povolen, prověříme následujícím příkazem, který vrátí tabulku obsahující přehled povolených procedurálních jazyků nad námi zadanou databází:

```
$ createlang -l asterisk

Procedural Languages
  Name   | Trusted?
---------+----------
 plpgsql | yes
```

A.2 Důležité příkazy pro práci s PostgreSQL

Připojení do databáze provedeme následujícím příkazem:

```
$ su postgres
$ psql <nazev databaze>
```

Následující příkaz vypíše tabulku existujících databází:

```
postgres=# \l
```

```
                                  List of databases
   Name    |  Owner   | Encoding |  Collation  |    Ctype    |   Access privileges
-----------+----------+----------+-------------+-------------+-----------------------
 asterisk  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 postgres  | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 |
 template0 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
                                                             : postgres=CTc/postgres
 template1 | postgres | UTF8     | en_US.UTF-8 | en_US.UTF-8 | =c/postgres
                                                             : postgres=CTc/postgres
(4 rows)
```

Důležitou součástí PostgreSQL jsou práva. Po vytvoření každé tabulky je nezbytnou součástí zadat jednotlivým uživatelům práva pro přístup. Syntaxe vypadá následovně:

```
GRANT <příkaz> ON <tabulka> TO <uživatel>
```

Práva pro dotaz na tabulku billing bude vypadat pro uživatele asterisk takto:

```
postgres=# GRANT SELECT ON billing TO asterisk;
```

Pokud je potřeba uživateli asterisk povolit plná práva k tabulce billing, použijeme příkaz:

```
postgres=# GRANT SELECT ON billing TO asterisk;
```

Pro ověření lze provést výpis práv pro tabulku billing:

```
postgres=# \dp billing
```

```
                              Access privileges
 Schema |  Name   | Type  |     Access privileges     | Column access privileges
--------+---------+-------+---------------------------+--------------------------
 public | billing | table | postgres=arwdDxt/postgres |
                          : asterisk=arwdDxt/postgres
(1 row)
```

Zdroj: [19],[20]

B Struktura databázových tabulek CDR a CEL

B.1 Tabulka CDR

Následující zdrojový kód pro vytvoření tabulky cdr byl převzat z [21].

```
CREATE TABLE cdr (
        calldate  timestamp NOT NULL ,
        clid  varchar (80) NOT NULL ,
        src  varchar (80) NOT NULL ,
        dst  varchar (80) NOT NULL ,
        dcontext varchar (80) NOT NULL ,
        channel varchar (80) NOT NULL ,
        dstchannel varchar (80) NOT NULL ,
        lastapp varchar (80) NOT NULL ,
        lastdata  varchar (80) NOT NULL ,
        duration  int  NOT NULL ,
        billsec  int  NOT NULL ,
        disposition  varchar (45) NOT NULL ,
        amaflags int NOT NULL ,
        accountcode varchar (20) NOT NULL ,
        uniqueid varchar (150) NOT NULL ,
        userfield  varchar (255) NOT NULL
);
```

Výpis 11: Vytvoření tabulky cdr

B.2 Tabulka CEL

Následující zdrojový kód pro vytvoření tabulky cel byl převzat z [22].

```
CREATE TABLE cel (
        id  serial  ,
        eventtype varchar (30) NOT NULL ,
        eventtime timestamp NOT NULL ,
        userdeftype varchar(255) NOT NULL ,
        cid_name varchar (80) NOT NULL ,
        cid_num varchar (80) NOT NULL ,
        cid_ani  varchar (80) NOT NULL ,
        cid_rdnis  varchar (80) NOT NULL ,
        cid_dnid  varchar (80) NOT NULL ,
        exten varchar (80) NOT NULL ,
        context  varchar (80) NOT NULL ,
        channame varchar (80) NOT NULL ,
        appname varchar (80) NOT NULL ,
        appdata varchar (80) NOT NULL ,
        amaflags int NOT NULL ,
        accountcode varchar (20) NOT NULL ,
        peeraccount varchar (20) NOT NULL ,
        uniqueid varchar (150) NOT NULL ,
        linkedid  varchar (150) NOT NULL ,
        userfield  varchar (255) NOT NULL ,
        peer varchar (80) NOT NULL
);
```

Výpis 12: Vytvoření tabulky cel

C Příloha na CD/DVD

Přiložené CD/DVD obsahuje následující soubory:

- create.sql
- data.sql
- t_billing.sql
- t_billing_cost.sql
- t_alert.sql
- fraud1.csv
- fraud2.csv